WEST Search History

Hide Items | Restore | Clear | Cancel

DATE: Friday, September 03, 2004

Hide?	Set Name	Query	Hit Count
		DB=USPT; PLUR=YES; OP=ADJ	
☐	L36	L35 and (cop$4 or duplicat$4) and (index$ same (url or universal resource locator))	3
☐	L35	l24 and l2	30
☐	L34	L33	0
☐	L33	L22 and (fram$2 or sub-frame$ or subframe$) and (category adj2 server$) and (stop$4 or freez$ or prohibit$4)	0
☐	L32	L22 and (fram$2 or sub-frame$ or subframe$) and (category adj2 server$) and (stop$4 or freez$ or prohibit$4) and (fram$2 or sub-frame$ or subframe$)	0
☐	L31	L22 and (fram$2 or sub-frame$ or subframe$) and (category adj2 server$) and 709/203.ccls.	0
☐	L30	L22 and (fram$2 or sub-frame$ or subframe$) and (category adj2 server$)	0
		DB=JPAB; PLUR=YES; OP=ADJ	
☐	L29	L24 and (cop$4 or duplicat$4)	0
☐	L28	L24 and 709/2$$.ccls. and (cop$4 or duplicat$4)	0
		DB=USPT; PLUR=YES; OP=ADJ	
☐	L27	L24 and 709/2$$.ccls. and (cop$4 or duplicat$4)	16
☐	L26	L24 and 709/2$$.ccls. and (cop$4 or duplicat$4)	16
☐	L25	L24 and 709/2$$.ccls.	16
☐	L24	L22 and (fram$2 or sub-frame$ or subframe$)	30
☐	L23	L22 and fram$2 and (sub-frame$ or subframe$)	1
☐	L22	L11 and (database or data base or data-base) and (network adj3 resource$) and (stop$4 or freez$ or prohibit$4)	32
☐	L21	L12 and (database or data base or data-base) and (network adj3 resource$) and (stop$4 or freez$ or prohibit$4)	0
☐	L20	L12 and (database or data base or data-base) and (network adj3 resource$) and (stop$4 or freez$ or prohibit$4)	0
☐	L19	L12 and (database or data base or data-base) and (network adj3 resource$)	4
☐	L18	(url or universal resource locator) same (stor$4 or locat$4) same (www or world wide web) adj2 page$ and (categor$2 near3 server)	2
		DB=PGPB; PLUR=YES; OP=ADJ	
☐	L17	(url or universal resource locator) same (stor$4 or locat$4) same (www or world wide web) adj2 page$ and (categor$2 near3 server)	2
		DB=EPAB; PLUR=YES; OP=ADJ	
		(url or universal resource locator) same (stor$4 or locat$4) same (www or	

h e b b cg b chh e c f ff e ch e h

☐	L16	world wide web) adj2 page$ and (categor$2 near3 server)	0
		DB=USOC; PLUR=YES; OP=ADJ	
☐	L15	(url or universal resource locator) same (stor$4 or locat$4) same (www or world wide web) adj2 page$ and (categor$2 near3 server)	0
		DB=EPAB; PLUR=YES; OP=ADJ	
☐	L14	(url or universal resource locator) same (stor$4 or locat$4) same (www or world wide web) adj2 page$	3
		DB=TDBD; PLUR=YES; OP=ADJ	
☐	L13	(url or universal resource locator) same (stor$4 or locat$4) same (www or world wide web) adj2 page$	1
		DB=USPT; PLUR=YES; OP=ADJ	
☐	L12	709/2$$.ccls. and l7	33
☐	L11	707/10$.ccls. and l1 and l2	431
☐	L10	(cop$4 or duplicat$4) and L9	8
☐	L9	(index$ same (url or universal resource locator)) and L7	10
☐	L8	l4 and L7	0
☐	L7	(categor$2 near3 server) and l1	74
☐	L6	(catergor$2 near3 server) and l1	0
☐	L5	(catergor$2 near3 server) and l3	0
☐	L4	(url or universal resource locator) same (stor$4 or locat$4) same (www or world wide web) adj2 page$	165
☐	L3	(creat$ or manipulat$4 or maintain$4) same (categor$4 or profile$) and (www or world wide web) adj2 page$	208
☐	L2	www or world wide web	13242
☐	L1	(creat$ or manipulat$4 or maintain$4) same (categor$4 or profile$)	57054

END OF SEARCH HISTORY

h e b b cg b chh e c f ff e ch e h

First Hit Previous Doc Next Doc Go to Doc#

End of Result Set

Generate Collection Print

L13: Entry 1 of 1 File: TDBD Jan 1, 1998

TDB-ACC-NO: NN9801711

DISCLOSURE TITLE: Information Retrieval and Presentation Apparatus with Version Control

PUBLICATION-DATA:
IBM Technical Disclosure Bulletin, January 1998, US

VOLUME NUMBER: 41
ISSUE NUMBER: 1
PAGE NUMBER: 711 - 712

PUBLICATION-DATE: January 1, 1998 (19980101)

CROSS REFERENCE: 0018-8689-41-1-711

DISCLOSURE TEXT:

Disclosed is a system for maintaining versions of information sources and for supporting temporal information retrieval and visualization of the information sources. Information sources could be a World Wide Web (WWW) page, a channel (in the sense of webcasting and push technology), or an output of an Internet search engine for a specified query. The disclosed system consists of three components: o Version controller o Version-based information extractor o Client manager The version controller retrieves and stores snapshots (versions) of information sources at a predefined update frequency (e.g., daily, every N days, weekly). The maximum number, K, of stored versions and the depth, D, of links for traversing referenced objects can also be specified for each information source. The version control mechanism can be one of existing full-version mechanisms, such as difference calculation and update sequences, but partial information extraction method, where specified segments (e.g., title and headers, or HTML anchors ...) are only extracted, can be used to maintain essential information of versions. This method may not be able to recover complete versions, but can drastically reduce the required memory space for storing versions. Irrelevant information, such as JAVA applets and style sheet specifications, can also be omitted from the versions. The updates (and, therefore, versions) between the specified sampling intervals may be totally ignored. A trigger for storing a version can also be a user's explicit operation of browsing a WWW page. By coupling a timestamp T (or a version number) with each URL U, a WWW browser (or a server and a proxy) can store a new version whenever a user accesses the URL (at timestamp T+) whose contents have been updated since T. Even though the contents may have not been updated, the version controller can store the information that "the URL U unchanged at timestamp T+" for finer version control. The version-based information extractor calculates the following information from a series of versions of information sources:

o Data items included in multiple versions of an information source o Data items included in only one version of an information source o Keywords (or phrases) that appear in one or more versions of an information source o Keywords (or phrases) that appear in multiple information sources o Keywords (or phrases) that appear in only one information source A data item could be ... or ... fillers, text in image

captions, contiguous paragraphs, etc. Given a query or a search profile, the information extractor can also calculate the following information using the above extracted features: o The top K information sources that have the largest number (or most frequently) of versions matching the given query/profile. o The top K information sources that have the latest N versions matching the given query/profile. o The top K information sources that have the longest series of versions matching the given query/profile.

The client manager provides the following functions: o temporal query: the client manager allows the following types of query: - Get the version of URL U around timestamp T - Get the version of URL U including the phrase P - Get the version of URL U when I found the phrase P in a version of URL V. A simple regular expressions can also be used to specify the candidate versions. For example, "http://.*/foo.html?Date=1997"56 .*" represents versions of a file foo.html in either May or June of 1997. Here, ".*" stands for any sequence of zero or more characters, and "56 matches any one of character inside ".* . o version list: the client manager shows versions of information sources in a variety of ways, including the timestamp ordering, size ordering, and ordering by the number of links. o linked-object recovery: If two WWW pages, A.html and B.html, have several versions at several timestamps (TA(1), TA(2), ..., and TB(1), TB(2), ...) and there is a link from A.html to B.html, the client evaluates the link from A.html at TA (i), and returns B.html at TB(j) such that TA(i) < TB(j), and there is no B.html at TB(k) satisfying TA(i) < TB(k) and TB(k) < TB(j).

o visualization: A series of versions of an information source can be used to visualize and better understand the dynamic aspects of the information source. Instead of browsing a specific snapshot, the series of versions can show: - how long each data item have been included - how much and how often the data items have been updated - what kind of topics and areas are mainly covered in the information source by using 2-dimensional and 3-dimensional graphic representations.

The disclosed system can be implemented in any of the three forms: (1) an information server at a host computer, (2) a client facility at a user's computer, and (3) a proxy enhancement of a computer connecting servers and clients.

Previous Doc Next Doc Go to Doc#

Hit List

Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs | Generate OACS

Search Results - Record(s) 1 through 8 of 8 returned.

☐ 1. Document ID: US 6785671 B1

L10: Entry 1 of 8 File: USPT Aug 31, 2004

US-PAT-NO: 6785671
DOCUMENT-IDENTIFIER: US 6785671 B1

TITLE: System and method for locating web-based product offerings

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWMC | Draw De

☐ 2. Document ID: US 6782350 B1

L10: Entry 2 of 8 File: USPT Aug 24, 2004

US-PAT-NO: 6782350
DOCUMENT-IDENTIFIER: US 6782350 B1

TITLE: Method and apparatus for managing resources

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWMC | Draw De

☐ 3. Document ID: US 6674453 B1

L10: Entry 3 of 8 File: USPT Jan 6, 2004

US-PAT-NO: 6674453
DOCUMENT-IDENTIFIER: US 6674453 B1

TITLE: Service portal for links separated from Web content

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWMC | Draw De

☐ 4. Document ID: US 6670968 B1

L10: Entry 4 of 8 File: USPT Dec 30, 2003

US-PAT-NO: 6670968
DOCUMENT-IDENTIFIER: US 6670968 B1

TITLE: System and method for displaying and navigating links

h e b b g e e e f e c ef b e

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWMC | Draw De

☐ 5. Document ID: US 6526442 B1

L10: Entry 5 of 8 File: USPT Feb 25, 2003

US-PAT-NO: 6526442
DOCUMENT-IDENTIFIER: US 6526442 B1

TITLE: Programmable operational system for managing devices participating in a network

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWMC | Draw De

☐ 6. Document ID: US 6438580 B1

L10: Entry 6 of 8 File: USPT Aug 20, 2002

US-PAT-NO: 6438580
DOCUMENT-IDENTIFIER: US 6438580 B1

TITLE: System and method for an interactive knowledgebase

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWMC | Draw De

☐ 7. Document ID: US 6131118 A

L10: Entry 7 of 8 File: USPT Oct 10, 2000

US-PAT-NO: 6131118
DOCUMENT-IDENTIFIER: US 6131118 A

TITLE: Flexible display of management data in a programmable event driven processing system

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWMC | Draw De

☐ 8. Document ID: US 6081840 A

L10: Entry 8 of 8 File: USPT Jun 27, 2000

US-PAT-NO: 6081840
DOCUMENT-IDENTIFIER: US 6081840 A

TITLE: Two-level content distribution system

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWMC | Draw De

h e b b g e e e f e c ef b e

Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs | Generate OACS

Term	Documents
COP$4	0
COP	3831
COPA	296
COPAA	1
COPAANE	4
COPAB	1
COPABIA	1
COPABLE	10
COPAC	5
COPACK	6
COPACO	3
((COP$4 OR DUPLICAT$4) AND L9).USPT.	8

There are more results than shown above. Click here to view the entire set.

Display Format: TI Change Format

Previous Page Next Page Go to Doc#

Hit List

Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs | Generate OACS

Search Results - Record(s) 1 through 4 of 4 returned.

☐ 1. Document ID: US 6662231 B1

L19: Entry 1 of 4 File: USPT Dec 9, 2003

US-PAT-NO: 6662231
DOCUMENT-IDENTIFIER: US 6662231 B1
** See image for Certificate of Correction **

TITLE: Method and system for subscriber-based audio service over a communication network

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw De

☐ 2. Document ID: US 6658463 B1

L19: Entry 2 of 4 File: USPT Dec 2, 2003

US-PAT-NO: 6658463
DOCUMENT-IDENTIFIER: US 6658463 B1

TITLE: Satellite multicast performance enhancing multicast HTTP proxy system and method

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw De

☐ 3. Document ID: US 6101527 A

L19: Entry 3 of 4 File: USPT Aug 8, 2000

US-PAT-NO: 6101527
DOCUMENT-IDENTIFIER: US 6101527 A

TITLE: System for managing and processing distributed object transactions and process implemented by said system

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw De

☐ 4. Document ID: US 6098108 A

L19: Entry 4 of 4 File: USPT Aug 1, 2000

US-PAT-NO: 6098108

h e b b g e e e f e c ef b e

DOCUMENT-IDENTIFIER: US 6098108 A

TITLE: Distributed directory for enhanced network communication

Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs | Generate OACS

Term	Documents
DATABASE	75830
DATABASIS	0
DATABASES	26997
DATA	860580
DATUM	15117
BASE	1113104
BASIS	535744
BASES	186236
DATA-BASE	660
DATA-BASIS	1
DATA-BASES	94
(L12 AND (DATABASE OR DATA BASE OR DATA-BASE) AND (NETWORK ADJ3 RESOURCE$)).USPT.	4

There are more results than shown above. Click here to view the entire set.

Display Format: TI Change Format

Previous Page Next Page Go to Doc#

h e b b g e e e f e c ef b e

L19: Entry 3 of 4 File: USPT Aug 8, 2000

DOCUMENT-IDENTIFIER: US 6101527 A
TITLE: System for managing and processing distributed object transactions and process implemented by said system

Abstract Text (1):
The present invention relates to a system and process for managing and processing object transactions in a network of distributed resources operating in the client-server mode, wherein the client sends a request to at least one transaction object contained in at least one of the servers (RS1, RS2, etc.) distributed across the network, while a transaction manager dialogues with a resource manager (RM) through a predefined interface by means of a transaction validation protocol. This system is noteworthy in that it achieves the implicit integration of resource managers (RM) adapted to the predefined interface, so as to integrate the participation of existing or future resource managers (RM) into a distributed transaction managed by the transaction manager, by providing objects capable of participating in the transaction validation protocol implemented by the transaction manager, which objects address the resource managers through the predefined interface. For this purpose, in the present system, each server comprises a specific local component (LOC1, LOC2, etc.) which encapsulates the calls to the predefined interface in the form of objects called resource objects (RSO), while moreover one server for managing the predefined interface (XAMS) is provided per domain for implementing the encapsulation of the transaction validation protocol, thus allowing multiple distributed objects to execute multiple requests in the same single transaction.

Brief Summary Text (3):
The present invention relates to a system for managing and processing object transactions in a network of distributed resources operating in the client-server mode, wherein the client sends a request to at least one transaction object contained in at least one of the servers distributed across the network, while a transaction manager dialogues with a resource manager through a predefined interface and by means of a transaction validation protocol. It also relates to the process implemented by this system.

Brief Summary Text (5):
Traditionally, and for a long time, the control and management of transaction data have been carried out by means of centralized mainframe computers. These machines were initially accessible locally, then later through networks which became increasingly complex, but were still hierarchical. It is only more recently that the distributed model based on open systems has been used. Generally, a distributed management environment makes it possible to integrate the administration of systems, networks and user applications, the dialogue between the various machines of the system and/or between the various users being organized around requests and responses to these requests, the most common requests in a network being related to file access or data access. An application is said to be designed according to a "client-server" architecture when it is comprised of two independent programs which cooperate with one another to implement the same process, each of which runs in an environment of its own (machine, operating system), and a programming interface using a language composed of commands makes it possible to control their dialogue. The client-server mode has the advantage of enabling a user (for example a simple microcomputer) called a client to delegate part of its task or its operations to be

executed to a server. In this way, the client has at its disposal a computing capacity much greater than that of its own microcomputer. Likewise, a client can address a specialized server and effectively outsource an operation, the server being under optimal conditions in terms of implementation and expertise due to its specialization. In this context, the object of the transaction processing service is to provide the specific functions required for running applications which modify a given situation in real time. Transaction processing applications use services which guarantee that the transactions are carried out completely or are not executed at all. It must be recalled here that a transaction is a set of commands which have no significance unless all of them are executed, a concept which guarantees the consistency and the integrity of the data. The completion of transactions, which is known as validation or consolidation ("commitment" to one skilled in the art), must consequently have certain characteristics. Thus, the transaction processing service must ensure that the transaction applications are "Atomic," "Consistent," "Isolated" and "Durable" (ACID), "atomic" meaning that all the elements of the transaction are processed or no element is processed, "coherent" meaning that if any part of the transaction is not executed, all the parts of the system affected by this transaction remain in their original state, "isolated" meaning that during the processing of a transaction, the shared resources of the system are not accessible to another transaction, "durable" meaning that the results of a completed transaction are permanent and are not lost, and thus that in case of a fault or failure, the transaction is not lost. All of these properties consequently make it possible to keep the data, which constitute a non-negligible part of the property of an organization, consistent and constantly updated no matter what type of failure occurs (program, system, hardware, or communications). Providing these properties has become more difficult as the transaction systems themselves have become more sophisticated. At the present time, transactions generally involve a plurality of systems and affect various data bases as well as various types of resources. In order to manage these systems, the transaction processing service manages the resources so as to guarantee the coordination of the validations and provides specialized communications for managing the distributed transaction processing applications. The transaction processing service must also coordinate the various applications which must be involved in processing a global transaction. For this reason, the transaction environment must offer a certain flexibility, the "X/OPEN" environment being a good example in that it makes it possible to effectively complement the transaction processing services working on large volumes of transactions and to manage an architecture using distributed transaction processing. In this way, the applications can use distributed data bases in a transparent manner.

Brief Summary Text (8):
More particularly, in this transaction context, the "X/OPEN" distributed transaction processing model defines resource managers as being components which authorize access to shared resources such as data bases, file systems or print servers. A resource wherein the data are assigned by the validation ("commit") or cancellation ("rollback") of a transaction is said to be recoverable. In the case of a validation ("commit"), the modifications and updates already executed are rendered effective, in the case of a cancellation ("rollback"), the resource remains in its original state before the transaction, and in case of error, the operations of the transaction in progress are cancelled. By controlling access to a recoverable shared resource, the resource manager makes it possible to guarantee that this resource will return to a consistent state after any potential failure. X/OPEN defines an interface, the XA interface, between the transaction manager and the resource manager. This predefined and standardized interface allows the involvement and the cooperation of heterogeneous resource managers in a single distributed transaction and adheres to a selected two-phase commit protocol managed by the transaction manager. The main interactions in the X/OPEN distributed transaction processing model are the following. First, a client application initiates a transaction, then by sending requests, it involves shared resources to which the transaction relates, which shared resources are managed by resource

managers. Next, the client application starts and finishes the transaction. At the completion of the transaction, the transaction manager contacts the resource servers and coordinates the two-phase commit protocol through the XA interface. However, a technological choice of this type has considerable drawbacks, particularly due to the complexity of the utilization of this interface by a server in this transaction environment. In effect, in order to execute a transaction, a server intending to use a data base must use this interface from the start to indicate its participation in this transaction and its intention to use the data base, and each time requests arrive, it must specify that the latter are part of the transaction, after which it completes its participation in the transaction, a response to the requests having been provided, while moreover, it must be able to communicate with the transaction manager so as to be capable of reacting to its promptings during the implementation of the two-phase commit protocol. This dialogue with the predefined interface becomes even more complex the larger the number of data bases this server intends to access. Moreover, the requests passing through the ORB only add to the complexity of a utilization of this type. Finally, the compatibility between relational data base systems and the specifications of the XA interface is far from complete, rendering any true portability problematic.

Brief Summary Text (12):
Thus, when a server uses a resource through a resource manager (for example a data base), the participation of this data base in the transaction is managed directly by the present system, which means that this data base is encapsulated, and all the calls to the predefined interface are executed implicitly and are hidden from the programmer of the application, who consequently does not have to worry about them.

Brief Summary Text (13):
Advantageously, in order for several servers to be able to use the same data base, each server comprises a specific local component which encapsulates the calls to the predefined interface in the form of objects called resource objects, while moreover one server for managing the predefined interface is provided per domain for implementing the encapsulation of the transaction validation protocol, thus allowing multiple distributed objects to execute multiple requests in the same single transaction.

Detailed Description Text (2):
A few reminders regarding the "X/OPEN" distributed transaction processing model will be useful at this point in order to better understand how the resource managers in conformity with this model can be integrated into applications based on the model of the present system according to the invention. The architecture of the present system allows an implicit integration wherein the participation of the resource managers in a distributed transaction is encapsulated into objects provided by this present system which control the transaction validation protocol through the XA interface. The "X/OPEN" distributed transaction processing model defines resource managers which, as indicated above, are components which authorize access to shared resources such as data bases, file systems or print servers. A resource is said to be recoverable when, having been allocated by a transaction, it can be modified if the transaction is validated ("commit") or remain in its original state if the transaction is cancelled ("rollback"). By controlling access to a shared recoverable resource, the resource manager makes it possible to guarantee that this resource will return to a consistent state after any potential fault or failure. X/OPEN also defines the XA interface between the transaction manager and the resource manager. This XA interface allows the involvement and the cooperation of heterogeneous resource managers in a single distributed transaction and adheres to a two-phase commit protocol managed by the transaction manager. A few instructions or routines used by this XA interface are explained below, it being understood that two types of routines are used. A first type allows a resource manager to call a transaction manager, a transaction manager in conformity with the X/OPEN model being assumed to provide the routines of this first type for allowing the resource managers to dynamically control their participation during

the initiation of transactions, thus:

Detailed Description Text (37):
For this purpose, two types of components make it possible to encapsulate the integration of these resource managers. The first type corresponds to a local component LOC1, LOC2, etc., installed in each recoverable server RS1, RS2, etc., the library of the transaction service being linked to each recoverable server accessing a resource manager RM. This library uses local components of the system according to the invention which make it possible to achieve the implicit association of the execution units relative to the transactions, and it is also this library which makes it possible to achieve indirect context management, and implicit support of propagation and control. The second type of component corresponds to a server XAMS for managing the predefined interface, in the present example the XA interface, which server XAMS manages resource objects capable of participating in the two-phase validation or commit protocol by encapsulating the calls to the predefined interface and registering the resource objects, one per transaction, through the coordination objects of the transactions. There is one server XAMS per data base domain which manages all the resource objects for the transactions associated with the recoverable servers for this given domain, a data base in this relational context being defined as the set of tables that a given user can access.

Detailed Description Text (39):
at the opening of the resource managers, certain data bases allow only one of their domains to be accessed per process, hence only one call xa.sub.-- open per process for a determined data base, and the resource managers are only initialized (xa.sub.-- open) during the operations for creating the local components of the system.

Detailed Description Text (47):
When any failures are discovered, automatic restart procedures are implemented. In the case of a failure involving the restart of the transaction management server, the restart procedure makes it possible to complete an interrupted transaction as long as the failure occurred after the recording of a validation decision. In this case, the transaction manger is capable of executing the second phase of the commit protocol, all other cases resulting in the cancellation of the transaction. In the case of a failure involving the restart of the server for managing the predefined interface, the restart procedure is comprised of contacting the resource manager, for example the data base, and of recovering the identifiers of the transactions that are in the prepared state (operation xa.sub.-- recover), then of recreating the corresponding resource objects so that the latter contact the transaction manager to indicate to it that they are ready to participate in the second phase of the commit protocol for the interrupted transaction.

Detailed Description Text (52):
In summary, a developer supplying transaction objects who wants the permanent, and therefore durable, data of his objects to be managed by a resource manager, for example a data base, must:

Detailed Description Text (56):
For this reason, the system according to the invention is mainly comprised of servers implementing the various objects used and of libraries which must be linked to the components of the client and/or server applications. Two categories of servers implement the various objects used. The first category corresponds to the transaction management server, which implements objects providing the functionalities of the system for creating transactions, coordinating the transaction validation protocol, for example a two-phase commit, completing and terminating transactions and coordinating recovery. The second category corresponds to the servers for managing the predefined interface, for example the XA interface, which manage resource objects capable of participating in the validation protocol

by encapsulating calls to the predefined interface, objects which achieve the implicit integration of the resource managers adapted to the XA interface, the latter being used only by applications using resource managers adapted to the XA interface to store permanent data related to transaction objects. As for the libraries, they implement the local components capable of providing indirect context management, and the control and management of local information related to the transaction context. Two types of libraries are provided, depending on whether or not the application uses resource managers adapted to the XA interface, these two types being implemented from shared libraries and being used in the direct context management and implicit transaction propagation modes. The first type of library contains the "stubs" (as they are known to one skilled in the art) generated by the interface definition language (IDL) compiler for accessing the objects of the transaction manager. The second type of library contains the stubs generated by the interface definition language compiler for accessing the objects of the server for managing the interface XA, this latter type being used only by applications using resource managers adapted to the interface XA to store data related to permanent objects.

Detailed Description Text (57):
The system according to the invention, due to its design, allows easy interconnection with any resource manager adapted to the XA interface for a large number of applications which use and access relational data base management systems such as, for example, Oracle (trademark of Oracle Corporation and Bull S.A.), Sybase (trademark of Sybase, Inc.), Informix (trademark of Informix Software, Inc.) etc.

Current US Original Classification (1):
709/201

Current US Cross Reference Classification (2):
709/221

CLAIMS:

1. A system for managing and processing object transactions in a network of distributed resources operating in the client-server mode, wherein the client sends a request to at least one transaction object contained in at least one of multiple servers distributed across the network, while a transaction manager dialogues with a resource manager through a predefined interface by means of a transaction validation protocol, comprising:

said system being arranged and configured to achieve implicit integration of resource managers adapted to the predefined interface so as to integrate participation of resource managers into a distributed transaction managed by the transaction manager, by providing objects capable of participating in the transaction validation protocol implemented by the transaction manager, which objects address the resource managers through the predefined interface;

each said server comprising a specific local component which encapsulates calls to the predefined interface in the form of resource objects, and one of said servers being designated for managing the predefined interface is provided per domain for implementing encapsulation of the transaction validation protocol, thereby allowing multiple distributed objects to execute multiple requests in the same single transaction.

Hit List

Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs | Generate OACS

Search Results - Record(s) 1 through 16 of 16 returned.

☐ 1. Document ID: US 6732181 B2

L25: Entry 1 of 16 File: USPT May 4, 2004

US-PAT-NO: 6732181
DOCUMENT-IDENTIFIER: US 6732181 B2

TITLE: Internet-enabled service management and authorization system and method

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw De

☐ 2. Document ID: US 6640244 B1

L25: Entry 2 of 16 File: USPT Oct 28, 2003

US-PAT-NO: 6640244
DOCUMENT-IDENTIFIER: US 6640244 B1

TITLE: Request batcher in a transaction services patterns environment

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw De

☐ 3. Document ID: US 6615253 B1

L25: Entry 3 of 16 File: USPT Sep 2, 2003

US-PAT-NO: 6615253
DOCUMENT-IDENTIFIER: US 6615253 B1
** See image for Certificate of Correction **

TITLE: Efficient server side data retrieval for execution of client side applications

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw De

☐ 4. Document ID: US 6611862 B2

L25: Entry 4 of 16 File: USPT Aug 26, 2003

US-PAT-NO: 6611862
DOCUMENT-IDENTIFIER: US 6611862 B2

h e b b g e e e f e c ef b e

TITLE: User station software that controls transport and presentation of content from a remote source

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw De

---

☐ 5. Document ID: US 6571282 B1

L25: Entry 5 of 16 File: USPT May 27, 2003

US-PAT-NO: 6571282
DOCUMENT-IDENTIFIER: US 6571282 B1

TITLE: Block-based communication in a communication services patterns environment

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw De

---

☐ 6. Document ID: US 6556659 B1

L25: Entry 6 of 16 File: USPT Apr 29, 2003

US-PAT-NO: 6556659
DOCUMENT-IDENTIFIER: US 6556659 B1
**** See image for Certificate of Correction ****

TITLE: Service level management in a hybrid network architecture

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw De

---

☐ 7. Document ID: US 6496850 B1

L25: Entry 7 of 16 File: USPT Dec 17, 2002

US-PAT-NO: 6496850
DOCUMENT-IDENTIFIER: US 6496850 B1

TITLE: Clean-up of orphaned server contexts

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw De

---

☐ 8. Document ID: US 6446076 B1

L25: Entry 8 of 16 File: USPT Sep 3, 2002

US-PAT-NO: 6446076
DOCUMENT-IDENTIFIER: US 6446076 B1

TITLE: Voice interactive web-based agent system responsive to a user location for prioritizing and formatting information

h e b b g e e e f e c ef b e

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw De

☐ 9. Document ID: US 6442547 B1

L25: Entry 9 of 16 File: USPT Aug 27, 2002

US-PAT-NO: 6442547
DOCUMENT-IDENTIFIER: US 6442547 B1
**** See image for Certificate of Correction ****

TITLE: System, method and article of manufacture for information service management in a hybrid communication system

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw De

☐ 10. Document ID: US 6434568 B1

L25: Entry 10 of 16 File: USPT Aug 13, 2002

US-PAT-NO: 6434568
DOCUMENT-IDENTIFIER: US 6434568 B1

TITLE: Information services patterns in a netcentric environment

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw De

☐ 11. Document ID: US 6401085 B1

L25: Entry 11 of 16 File: USPT Jun 4, 2002

US-PAT-NO: 6401085
DOCUMENT-IDENTIFIER: US 6401085 B1

TITLE: Mobile communication and computing system and method

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw De

☐ 12. Document ID: US 6356905 B1

L25: Entry 12 of 16 File: USPT Mar 12, 2002

US-PAT-NO: 6356905
DOCUMENT-IDENTIFIER: US 6356905 B1

TITLE: System, method and article of manufacture for mobile communication utilizing an interface support framework

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw De

h e b b g ee e f e c ef b e

☐ 13. Document ID: US 6199099 B1

L25: Entry 13 of 16 File: USPT Mar 6, 2001

US-PAT-NO: 6199099
DOCUMENT-IDENTIFIER: US 6199099 B1

TITLE: System, method and article of manufacture for a mobile communication network utilizing a distributed communication network

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw De

☐ 14. Document ID: US 6195697 B1

L25: Entry 14 of 16 File: USPT Feb 27, 2001

US-PAT-NO: 6195697
DOCUMENT-IDENTIFIER: US 6195697 B1

TITLE: System, method and article of manufacture for providing a customer interface in a hybrid network

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw De

☐ 15. Document ID: US 6134548 A

L25: Entry 15 of 16 File: USPT Oct 17, 2000

US-PAT-NO: 6134548
DOCUMENT-IDENTIFIER: US 6134548 A

TITLE: System, method and article of manufacture for advanced mobile bargain shopping

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw De

☐ 16. Document ID: US 5884035 A

L25: Entry 16 of 16 File: USPT Mar 16, 1999

US-PAT-NO: 5884035
DOCUMENT-IDENTIFIER: US 5884035 A

TITLE: Dynamic distributed group registry apparatus and method for collaboration and selective sharing of information

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw De

h e b b g e e e f e c ef b e

Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs | Generate OACS

Term	Documents
709/2$$	0
709/200	629
709/201	1062
709/202	715
709/203	2303
709/204	633
709/205	390
709/206	966
709/207	288
709/208	357
709/209	137
(L24 AND 709/2$$.CCLS.).USPT.	16

There are more results than shown above. Click here to view the entire set.

Display Format: TI Change Format

Previous Page Next Page Go to Doc#

h e b b g e e e f e c ef b e